System Integration

Microservices Design Principles

Patrick Stolc
M.Sc. Robotics Engineering

ERHVERVS AKADEMI SYDVEST

How do we design microservices?

💾 Design principles

- **Domain-Driven Design**
- **Single Responsibility**
- **Loose Coupling**

💾 Architectural principles

- **Data ownership**: Each microservice should own its own data.
- **Service boundaries**: Define clear boundaries between microservices.
- **Service interactions**: Define how microservices interact with each other.

Domain-Driven Design

💾 Definition

- A microservice should be designed around a specific domain or business area.

💾 Example

- In a e-commerce system, the **Product** microservice would handle product data and pricing.
- The **Order** microservice would handle order processing and fulfillment.
- The **Payment** microservice would handle payment processing.

Single Responsibility

💾 Definition

- Each microservice should have a single responsibility.

💾 Example

- A microservice that handles user authentication should only handle user authentication. It should not handle user data or user management.

Loose Coupling

💾 Definition

- Microservices should be loosely coupled, meaning they should have minimal dependencies on each other.

💾 Example

- A microservice that handles user authentication should not depend on a microservice that handles user data.